

NASA Space Station Software  
Standards Issues

George D. Tice, Jr.  
Tektronix, Inc.  
P.O. Box 4600 (M/S 92-525)  
Beaverton, Oregon 97075  
(503) 629-1310

ABSTRACT

The selection and application of software standards present the NASA Space Station Program with the opportunity to serve as a pacesetter for the United States software in the area of software standards. This presentation summarizes and discusses the strengths and weaknesses of each of the NASA defined software standards issues:

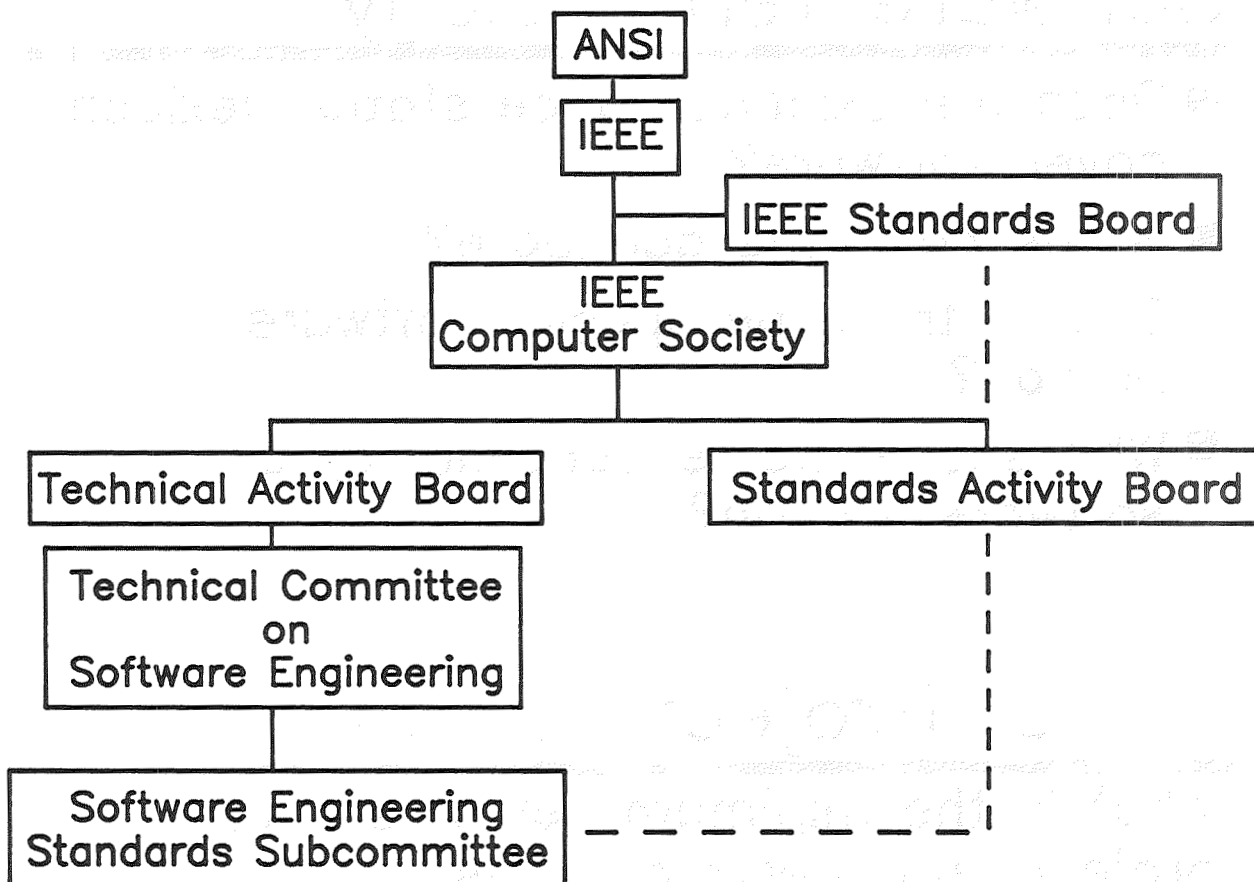
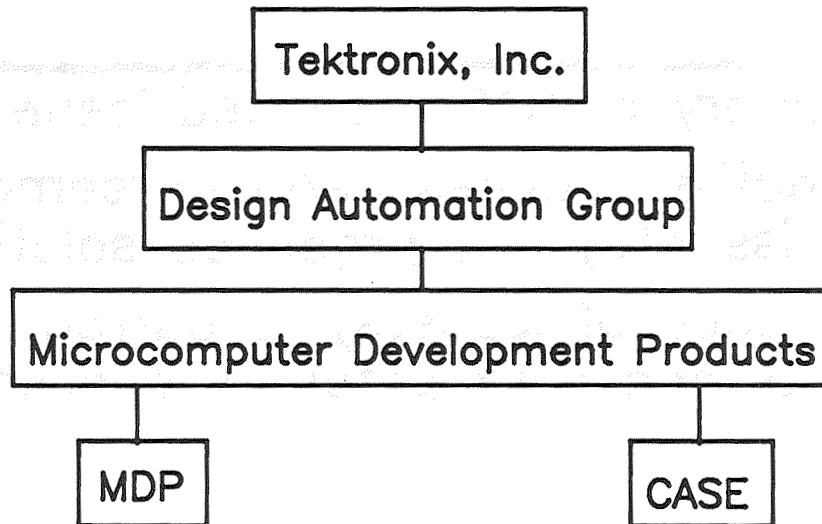
- Need for Common Software Terminology
- Project Directives
- Software Technology
- Software Portability
- Languages
- Documentation

Several additional significant standards issues are offered for NASA consideration:

- Value of Standards
- Potential Leverage from Other Standard Efforts

The presentation concludes with a challenge for the NASA Space Station Program to serve as a pacesetter for the U.S. Software Industry through:

- Management commitment to software standards
- Overall program participation in software standards
- Employment of the best available technology to support software standards



## Format

---

- Summary of NASA–defined issue(s)
- Strengths/weaknesses/disagreements with issue(s) and proposed solution(s)
- Relevance of issue(s) to current R&D efforts and their potential application

## Issue: Need for Common Software Terminology

---

- Does the existing space station lexicon cover software?
- Is the coverage adequate?
- Should there be special software lexicon?
- Who should be responsible for a software lexicon?

## Issue: Project Directives

---

What is the minimum set of software project practices/standards?

## Issue: Software Technology

- What criteria should be used to select SSIS software technology?
  - software engineering and practices
  - standards for portability
  - programming language
  - whether to impose an instruction set architecture
  - data driven vs. data embedded software
- What criteria should be used for technology changeover?
- How can we make technology change transparent?
- How do we keep current in technology?

## Issue: Software Portability

Applicability and methodology of portability and transferability for the space station.

## Issue: Languages

Languages for software development?

## Issue: Documentation

---

- What is the critical, minimal set of documentation and what level of detail should be specified?
- Do the critical set of documents and level of detail vary with software category?
- What acceptance criteria are needed?

## Additional Significant Standards Issues for NASA Consideration

### Issue: Value of Standards

---

- Education
- Simplification
- Conservation
- Certification
- Contribution

## Issue: Potential Leverage from Other Standard Efforts

---

- Department of Defense (DoD)
- European Space Agency (ESA)
- IEEE Software Engineering Standards

NASA

Space Station Program

Challenge  
and  
Opportunity

Serve as a Pacesetter for the  
U.S. Software Industry

---

- Management commitment to software standards
- Overall program participation in software standards
- Employment of the best available technology to support software standards



## DISCUSSION OF RECOMMENDATIONS

1. NASA Space Station management should establish policy supporting software standards which:

- A. States top level (levels A & B) endorsement and commitment.
- B. Defines implementation and enforcement authority and mechanism.
- C. Provides methodology for software standards training and encourages its use.
- D. Provides an overview (audit) program to measure effectivity and encourage adherence to software standards.
- E. Encourage technology infusion/insertion.

To be effective, standards must have top management's unconditional support and that support must be visible at all levels of activity. Unless the purpose of each standard is understood and the methodology for selecting, implementing and enforcing standards is known to be rational, they will be viewed with suspicion. It is necessary to continuously maintain the currency of software standards to ensure their utility, and thereby their continued use.

2. The NASA Space Station Program should establish a structure to develop and support software standards having the following characteristics:

- A. Level A management authorizes the structure to support software standards.
- B. A Space Station Software Standards Organization at level B with responsibility for promulgating, maintaining and enforcing software standards.
- C. A Software Standards Advisory Committee with level C representation to advise the Software Standards Organization on the need, feasibility and acceptance of proposed changes to software standards.

The mechanism for supporting software standards must be structured such that issues can be resolved at the appropriate levels. It must remain in constant touch with the user community to understand their requirements for and problems with software standards. It must be flexible enough to act quickly when change is needed and strong enough to resist change when that change will weaken the overall system of standards.

3. The NASA Space Station Program should proceed to acquire standards as follows:

- A. Establish a need for software standards based on Space Station system/software requirements.
- B. Establish a standard for standards to promote understandability and improve communication.
- C. Establish a priority for source selection of standards (international, ESA, industry, NASA, contractor, etc) that supports both the objectives and needs of the Program and organizations involved in it.

- D. Review existing software standards in light of requirements and priorities.
- E. Select and tailor from existing standards when possible and develop new standards as a last resort.
- F. Implement new standards on a trial basis with specific criteria for rejection and full implementation.

As with any system, it is critical that the most essential elements of the Space Station software standards system be identified early so that they may be brought into being in the proper sequence. This will enable the needed standards to be available at the appropriate time and avoid a bottom-up muddle of incompatibility.

4. The NASA Space Station Program should immediately act to satisfy its needs for software standards in the following areas:

- A. Common Software Terminology (Lexicon)
- B. Software Engineering Methodology and Practices
  - o Software Management
  - o Software Acquisition
  - o Software Development
  - o Coding
  - o Documentation
  - o Measurement and Data Collection
- C. Languages
- D. Instruction Set Architecture
- E. Networks
- F. Operating Systems
- G. Applications (e.g. DBMS)
- H. Security

These candidate areas represent the fundamental variables that must be managed and controlled through standardization to provide for a cost-effective software acquisition and support activity.



## CONCLUSIONS

Results of the panel deliberations were summarized at the close of the forum. Subsequently, the panel members carefully documented their findings and these contributions are included in this publication. The panels identified many issues and provided recommended actions for NASA to consider. These are now under study by members of the Space Station Program.

It is noteworthy that the recommendations from the panels are practical and workable, rather than academic long-range forecasts (e.g., go with Ada, identify a mandatory SDE, begin with Unix).

Although each panel operated independently, establishing its own format and leading its open forum discussions, there were several common themes that emerged, as noted by representatives of the Software Working Group who attended each of the panel open sessions. Some of the common topics mentioned by more than one panel were as follows.

- Many issues, especially in the management and standards areas, apply to systems rather than just to software. Software should be developed and managed as an integral part of a systems level strategy.
- Incremental development methodology should be practiced.
- Interfaces between software components and between hardware and software should be identified early and then managed.
- Technology evolution must be accommodated over the Space Station lifetime.
- Selection of computer hardware should not be a limiting factor on the software.
- NASA has much experience with large software projects and should use the lessons learned from the past in this development. Also, provision should be made to capture lessons learned during the development and operation of the Space Station for the benefit of future projects and the continued evolution and growth of the Space Station itself.
- Focus on maintenance, plan for it from the beginning.
- Begin training in software areas early (e.g., Ada programming, SDE use, software management procedures).

Some common concerns were also expressed during the various open sessions

- Schedule constraints indicate there is very little time for good "front-end" work.
- Many software terms are subject to individual interpretation and should be specifically defined, such as rapid prototyping, incremental development, users, languages and tools, risk management, life cycle, use of Ada, training. (Note that the proposed Space Station Software Lexicon will address this concern.)
- A potential incompatibility exists between "design-to-cost" and "life cycle costing", since it may not be possible to simultaneously optimize front-end and back-end costs.

## SOFTWARE MANAGEMENT PANEL CONCLUSIONS

The Software Management Panel agrees with NASA's assessment of the critical importance of software to the success of the Space Station Program. This is exemplified by the implementation of NASA-wide software cognizant organizations and support functions. The early production and substantial content of the draft Level A/B Software Management Plan are indicative of the ability and viability of the software organization. These are good beginnings.

The panel members have reviewed and deliberated on the Space Station Software Issues Report and the Software Management Plan. The resultant views were merged with the perceptions of industry and NASA invited representatives in open forum to produce a substantial and well-founded set of recommendations. These recommendations will facilitate further progress toward a meaningful, operative Software Management Plan. Given NASA's commitment to software excellence and the unique technical challenges of Space Station, the following conclusions are clear to the panel.

1. The necessary tasks and responsibilities envisioned for the Level A and B software management organizations far exceed their current and projected resources and authority. This is particularly the case at Level A. Resolution of this situation is fundamental to the success of the Space Station Program.
2. NASA needs to expand its excellence from in-house engineering to the arms-length acquisition of software in many categories to operate together in a very large system.
3. The focus of software management and acquisition should shift to maintenance/sustaining engineering in order to minimize life cycle costs.
4. Management procedures for interaction with non-Space Station services and users should be properly defined.
5. The scope of the top level software engineering and integration of Space Station software should be assessed and addressed.
6. The Software Management Plan should be restructured to recognize the foregoing needs.

## SOFTWARE DEVELOPMENT ENVIRONMENT PANEL CONCLUSIONS

The concept of a uniform SDE furnished and mandated by NASA, to address the critical life cycle cost and integration issues of Space Station software, is strongly endorsed. Risks, such as schedule, technological obsolescence, and contractor incompatibilities, can be mitigated by an incremental acquisition strategy, the use of layered architectures to assure technological transparency, and an operational concept which provides for contractor options to use their own SDEs, as long as the delivered software is supportable by the NASA SDE. This operational concept should be developed soon and should address user requirements and life cycle scenarios based on inputs from users, Phase B contractors, and similar DoD efforts (e.g., the JSSEE Operational Concept Document).

The SDE should focus on products (such as specifications, design/code representations, etc.) rather than specific methodologies, and should encourage the reuse of previously developed software in order to save costs.

The architecture of the SDE should be modularized and layered to allow for technological evolution at distinct levels. The operating system should be vendor and device independent insofar as possible. Unix appears to be the only candidate that meets these criteria, and should be considered as the initial basis for the operating system.

The SDE should be furnished as a portable software package (so that changes in hardware do not affect software functionality) and should consist of a subtable set of tools engineered with uniform interfaces providing the capability to customize to specific user requirements by application (e.g., flight or ground software development, analysis, management, simulation), by type of user (e.g., expert/novice, specialist/generalist), or by type of equipment (e.g., mainframe, mini, or work station). A major objective is to maximize commonality of widely used functions. The SDE should automatically collect data that characterize its use, and these data can be used as the basis for improvement and extension of the SDE.

#### LANGUAGES PANEL CONCLUSIONS

The selection and standardization of languages and interfaces for the Space Station program are critical needs to insure the success of this predominately engineering activity. While the Language Panel recognizes that the project life cycle will require a family of languages for the various classes of users and developers, it is crucial to begin making decisions which will focus planning efforts by limiting the range of possible selections. Requirements for the Space Station information system long-term maintenance and evolution will mandate that a high-order development language be utilized. It is recommended that the primary high-order language for source code development be Ada. Issues related to the utilization of Ada should be addressed as soon as possible. These include developing a transition strategy, providing education, accommodating the utilization of software already in existence, and developing fall-back options for high risk areas. One high-risk area is satisfying the requirements for run-time support for target systems, especially when the targets are distributed. Requirements for design specification languages or interfaces that complement Ada should be determined.

Additional conclusions are that the premature commitment to hardware implementation decisions should be avoided, and that the critical need is to develop the system and software architectures for Space Station.

#### SOFTWARE STANDARDS PANEL CONCLUSIONS

Standards are one of several elements that provide a common "backbone" for all software aspects of the Space Station Program. The operational Station will eventually influence NASA's entire future space activities. Because of this broad area of impact, the importance of software standardization for the Space Station Program must not be underestimated.

This forum has resulted in recommendations that encompass the major aspects of the needed software standards program. The procedure for arriving at these recommendations ensured that NASA received the best possible advice available in the area of software standards. Due to the farsightedness of the Space Station Software Working

Group, there is adequate, but not excessive, time available to implement the Standards Panel's recommendations.

A unique situation and opportunity has been created. The Space Station Program has received needed advice from experts at a key point in the Program on a critical subject. That subject happens to be software standards, an area of technology that has been stalled for too long a period. NASA is in a position not only to establish an outstanding software standards program for the Space Station, but to provide the software industry with a much needed innovative model in this area. NASA should move at once to act on the recommendations provided.

Additional standards issues should be addressed including the distributed network operating system, graphics (Core vs. GKS), standards for device independence (VDI, VDM for storing graphics, IGES or NAPLPS for transmission), program/operating system standard interface, self-documenting data record format, and OSI communications protocol standards.

#### REFERENCES

1. Voigt, S. J. (Ed.): Space Station Software Issues. NASA CP-2361, 1985.
2. Military Standard Defense System Software Development. DOD-STD-2167, 1985.
3. Reference Manual for the Ada Programming Language. ANSI/MIL-STD-1815A-1983, 1983.